

INTERACTIVE SOLUTION-ADAPTIVE GRID GENERATION

Yung K. Choo
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

Todd L. Henderson*
University of Illinois
Urbana, Illinois 61801

SUMMARY

TURBO-AD is an interactive solution-adaptive grid generation program under development. The program combines an interactive algebraic grid generation technique and a solution-adaptive grid generation technique into a single interactive solution-adaptive grid generation package. The control point form uses a sparse collection of control points to algebraically generate a field grid. This technique provides local grid control capability and is well suited to interactive work due to its speed and efficiency. A mapping from the physical domain to a parametric domain was used to improve difficulties that had been encountered near outwardly concave boundaries in the control point technique. Therefore, all grid modifications are performed on a unit square in the parametric domain, and the new adapted grid in the parametric domain is then mapped back to the physical domain. The grid adaptation is achieved by first adapting the control points to a numerical solution in the parametric domain using control sources obtained from flow properties. Then a new modified grid is generated from the adapted control net. This solution-adaptive grid generation process is efficient because the number of control points is much less than the number of grid points and the generation of a new grid from the adapted control net is an efficient algebraic process. TURBO-AD provides the user with both local and global grid controls.

INTRODUCTION

Quality grids are needed to obtain accurate numerical simulations of complex flows. But generating quality grids is difficult for complex problems because the grids must be provided without a prior knowledge of the flow details. Precise local and global grid controls are often difficult even when details of flow are known. A grid adaptation process (refs. 1 to 3) that couples the grid with the evolving flow solution can provide a means of obtaining proper distribution of grid points. This adaptation process is combined with the control point form of algebraic grid generation (refs. 4 to 6)

*Summer student intern at NASA Lewis Research Center.

to improve the efficiency of the adaptation process and to provide explicit local/global grid control capability.

TURBO-AD under development starts with a given initial grid and a given initial solution, and it redistributes the grid points to provide denser grid in high gradient areas of a solution while having sparse grid in low gradient areas. New flow solution obtained from the adapted grid is able to capture more significant flow features in the high gradient areas. The program runs on the IRIS 4D series workstations interactively. Interactive process of the code allows the user to immediately see the results and modify the grid as desired along the way.

In TURBO-AD, all grid modifications are performed in a parametric domain. The grid adaptation is obtained through adapting the control points to a source distribution, which is defined by the gradients of a numerical flow solution. The new adapted grid is then computed from the new adapted control points. The control point adaptation offers advantages over the direct grid adaptation. It increases the efficiency of the solution-adaptive procedure because the number of control points is significantly fewer than the number of grid points. It also provides smooth grids resulting from the control point form. The solution adaptation of a grid does not always result in a smooth grid for many adaptation techniques. But, when a nonsmooth set of control points is used to generate a grid the resulting grid is much smoother. The grid modifications in the square parametric domain offers another advantage. It allows the control points and grid points on the boundary to freely move with very little alteration of the given boundary geometry. It also prevents grids from tangling that can happen in concave regions with the control point formulation since there are no concave boundaries in the parametric domain. This paper briefly discusses the control point form of algebraic grid generation and solution-adaptive grid generation using parametric mappings. Then interactive solution-adaptive grid generation procedure is discussed with an example.

ALGEBRAIC CONTROL POINT GRID GENERATION

The control point form of algebraic grid generation and its application are presented in references 4 to 6 in detail and are not repeated in this paper. This section only contains a brief sketch of the control point form. The control point array is a sparse grid-type arrangement of locations in space with an index for each direction. A two-dimensional control point array, $(C_{i,j})$ is shown in figure 1. As an algebraic method, the control point form provides explicit control of the grid shape and spacing through the movement of the control points.

A fundamental part of the control point formulation is the construction of curves. This construction represents algebraic coordinate generation in a single direction between two opposing boundaries. These boundaries are the fixed endpoints of the curves. The remaining control points are in the interior of the sequence and are used to control the shape of the curve.

From any one control point, the two neighboring points on each side form local segments within the entire piecewise linear curve. The local segments about each control point define a change in

direction. The rate of change is determined by the two linear connections attached to the given control point as the curve assumes the respective directions between each pair of control points.

A continuous direction field is obtained in a smooth manner by interpolation. The independent variable for the interpolation is the curve parameterization. The interpolated result defines the field of vectors that are tangent to the desired curve. Simply stated it is an interpolation of first parametric derivatives. This determines a smooth first derivative of the entire curve. The desired curve is then obtained by a parametric integration. The integration here is taken so that the curve connects the specified endpoints.

To obtain local grid controls, local interpolation functions are used. With local functions, the movement of a control point results in an alteration of the constructed curve that is restricted to a local region about the point. The remaining regions are unaltered. The simplest local interpolants are the piecewise linear functions that do not vanish over at most two intervals. Example curves constructed from left to right boundaries are $E_2(\xi)$ and $E_j(\xi)$ in figure 1. Similarly, a curve $F_i(\eta)$, can be constructed for the other direction from bottom to top boundaries. The tensor product form, $T(\xi, \eta)$, depends only upon C_{ij} . The tensor product matches E_j or F_i at the extremities of i and j (i.e., at corner points in fig. 1 for instance).

When boundaries are to be specified, the corresponding data appear at the extremities of the values for ξ and η . Since the coordinate transformations are generally expressed in the form of a vector for the desired positions of all points in physical space, it is convenient to express the boundary specifications in terms of the position vector. Thus, the boundaries are denoted by $P(1, \eta)$, $P(N - 1, \eta)$, $P(\xi, 1)$, and $P(\xi, M - 1)$. To include the boundaries, the multisurface transformation is performed again as above, but now with the actual boundaries inserted. This results in a modification of T for the ξ and η directions, respectively. In each such directional construction, the actual boundaries appear as end conditions for the corresponding variable while the remaining boundaries are generated by the control points. Thus, by subtracting T from the sum of both directional constructions, the actual boundaries become end conditions for each variable. This process follows a Boolean sum format and upon simplification becomes

$$\begin{aligned} Q(\xi, \eta) = & T(\xi, \eta) + \gamma_1[1 - G_1(\xi)][P(1, \eta) - F_1(\eta)] \\ & + \gamma_2 G_{N-1}(\xi)[P(N - 1, \eta) - F_N(\eta)] \\ & + \gamma_3[1 - H_1(\eta)][P(\xi, 1) - E_1(\xi)] \\ & + \gamma_4 H_{M-1}(\eta)[P(\xi, M - 1) - E_M(\xi)] \end{aligned} \quad (1)$$

where G and H are integral of the local interpolation functions, and each of the four terms following the tensor product $T(\xi, \eta)$ represents a transfinite conformity to a boundary when each boundary on-off switch, γ , is 1. By setting any γ_i to 0, the corresponding boundary becomes available for free form modeling by means of the control points. In the order of appearance, the boundaries are for $\xi = 1$, $\xi = N - 1$, $\eta = 1$, and $\eta = M - 1$. Further details of the control point form are presented in references 4 and 5.

SOLUTION-ADAPTIVE CONTROL POINT GENERATION

The solution-adaptation of grids is usually accomplished by directly adapting all the grid points. This can be time consuming for many adaptation algorithms. To increase the grid adaptation speed, the procedure described in reference 1 is used to adapt the control net instead of the grid. This new approach can easily be more than ten (10) times faster in two-dimensional than the usual direct grid adaptation procedure of reference 1. The new grid generation from the adapted control net is an efficient algebraic process.

The adaptation procedure using the control points is illustrated in figure 2. It starts from the initial control net in the physical space (x,y) as shown in figure 2(a). The position of the initial control points is determined by the distribution of the initial grid points, from which the control net is constructed by "attachment" (ref. 5), and the definition of the local interpolation functions. In order to control the grid structure locally, local interpolation functions are chosen for TURBO-AD. The interpolation functions are nonzero over one interval about the first and last partition points on the opposite boundaries, while the interpolation functions over the internal partition points are nonzero over two intervals. Resulting from this is the first fine control net meshes around the boundary as seen in figure 2(a). The procedure then defines a control net in the parametric space (s,t) as shown in figure 2(b). This control net is mapped into another parametric space (u,v) to establish a uniform control net as shown in figure 2(c). The initial grid is also mapped into the parametric space (u,v) using the same procedure. Then two control-net control sources are defined at every grid point in u and v directions. The sources are defined by a linear combination of the first and second derivatives of one or more monitor functions, ϕ_n , in each parametric direction as shown in equations (2a) and (2b)

$$\sigma_{k\ell}^u = \sum_{n=1}^M \left\{ w_{1n}^u \left| \frac{\partial \phi_n}{\partial u} \right| + w_{2n}^u \left| \frac{\partial^2 \phi_n}{\partial u^2} \right| \right\} \quad (2a)$$

$$\sigma_{k\ell}^v = \sum_{n=1}^N \left\{ w_{1n}^v \left| \frac{\partial \phi_n}{\partial v} \right| + w_{2n}^v \left| \frac{\partial^2 \phi_n}{\partial v^2} \right| \right\} \quad (2b)$$

where k and ℓ are the indices of the grid point where the source is located. The w 's are weighting parameters that allow the user to control the contribution from each term. The monitor functions, ϕ , can be any flow property (i.e., density, pressure, Mach number, etc.).

The control point in the parametric domain, (u,v) , is now modified using equations (3a) and (3b).

$$u'_{ij} = u_{ij} + \sum_{k,\ell} K_{ijk\ell} \sigma_{k\ell} \quad (3a)$$

$$v'_{ij} = v_{ij} + \sum_{k,\ell} K_{ijk\ell} \sigma_{k\ell} \quad (3b)$$

where $K_{ijk\ell}$ is the influence coefficients. The influence coefficient controls the effects of a source at a grid location, (k,ℓ) , on a control point, (i,j) . The (u',v') domain is normalized to go from zero to one. The modified new control net is shown in figure 2(d). The computation of the summation in equation (3) can be very costly if the influence coefficient K has to control the effects of the source on all grid points instead of the control points. This will be evident in a sample interactive case as the number of control points is significantly less than grid points. The coefficients are defined by equations (4a) and (4b).

$$K^u_{ijk\ell} = \frac{u_{ij} - u_{k\ell}}{d} \exp(-\alpha_u d) \quad (4a)$$

$$K^v_{ijk\ell} = \frac{v_{ij} - v_{k\ell}}{d} \exp(-\alpha_v d) \quad (4b)$$

where

$$d = \sqrt{(u_{ij} - u_{k\ell})^2 + (v_{ij} - v_{k\ell})^2}$$

and the α 's are decay parameters for each of the parametric directions. The degree of adaptation is controlled by the user's input of the w 's and α 's.

In order to improve the efficiency, a cut-off value has been implemented such that the summation in equation (3) is terminated for sources where the $K_{ijk\ell}$'s decay below a user specified value. The input for this cut-off parameter, β , ranges between zero and one. At $\beta = 0$, the summation includes all points, and at $\beta = 1$, the summation would include no points.

Also, the sources are mirrored across each boundary to improve the orthogonality of the adapted control net to the boundary. Orthogonality of the control net lines can also be enforced by making the control net orthogonal to the boundary as described in reference 5.

The new control net points in the (u',v') domain have been repelled by the strong sources. This can be seen in figure 2(d). Now a new uniform control net is mapped back to the (s,t) domain using the mapping from the (u',v') domain to the (u,v) domain. This results in a clustering of points near the strong sources in the original parametric domain as shown in figure 2(e). At this point the new control net points are used to calculate a new parametric grid which is then mapped into the physical domain with the control net. The adapted control net in the physical domain is shown in figure 2(f).

INTERACTIVE SOLUTION-ADAPTIVE GRID GENERATION, SAMPLE CASE

The following example demonstrates main features available in TURBO-AD. The case to be considered is a ramp/expansion with an inflow Mach number of 2.0. The initial control net, grid, and Mach number distribution can be seen in figure 3. TURBO-AD uses the workstation mouse for the primary input. The right button either exits a current operation or displays the current menu. The menu selection is accomplished by highlighting the appropriate choice and then releasing the right button. The left and center mouse buttons are typically used for controls within the operations.

TURBO-AD can be executed by typing 'turboad.' The program then ask the user to type in the grid file name followed by the solution file name. The solution file can contain one to five scalar functions. The solution file for the sample case contains density, pressure, temperature, Mach number, and entropy, in that order. The solution file is optional. If a 'return' is entered at the request for a solution file, the user has access the interactive grid modification features discussed in reference 6 about TURBO-I.

Once TURBO-AD has been started, it will create the parametric mapping in memory and then display the initial grid and control net. The user can then bring up the home menu by pressing the right mouse button and holding it down. This menu is the top level or main menu. The user can select "MODIFY VIEW" which will take the user to the viewing control menu. Then by pressing the right mouse and holding it down, the user will see the "MODIFY VIEW" menu choices. The input function distribution can be displayed by selecting "Contour Function." This is a pull down menu, so the user should move the mouse to the right side of the selection. This will bring down a listing of functions that can be contoured. By selecting the fourth function the user can see the Mach number contour for the example case. It should be noted that the first function in the users's solution file is "function 1," the second is "function 2," etc. The "Scroll View" option allows the user to look at the control net alone, the grid alone, or many combinations. For this example, the user should see the control net, grid, and Mach number distribution if "Scroll View" is selected several times.

The user is now ready to prepare for the control net (CN) adaptation. The "ADAPTATION CONTROL" menu option can now be selected from the main menu or the "MODIFY VIEW" menu. Under the "ADAPTATION CONTROL" menu the user should use the mouse to pull down the "Active Functions" menu. This menu is to control which functions are used for adaptation. The default is for all functions to be turned off. The user can toggle the functions on or off by selecting the appropriate one. For this example the fourth function is selected. Please note that there is no correlation between the function being viewed and the functions selected under "Active Functions" for the CN adaptation.

The 'adapt.par' file contains the user definable parameters, w 's and α 's, shown in equations (2) and (4). The user is now ready to calculate the sources discussed in equations (2a) and (2b). This can be done by selecting the menu option "Calculate Source Terms." When the calculation of the sources is finished, the user will succeed in bringing up the menu. The calculation of the sources has now produced two new functions, σ'' and σ' , in the users list of functions. At this point, the user could view the source distribution using the built in color contouring available under "MODIFY VIEW." If the user contours the sixth function for this example case, using the 'adapt.par'

file, the user should see the source distribution in the s direction shown in figure 4. On the workstation screen, the blue colors show the lower source strength and the red show the higher source strength. The locations where high values exist is where the adapted control net lines should cluster. The source distribution in the t direction has been set to zero using the input file in this example. The user can now return to the adaptation menus.

The control net adaptation can now be started. To do this the user should select "Adapt CN" entry in the "ADAPTATION CONTROL" menu. This operation will take about 20 to 90 sec depending on which IRIS 4D series workstation is being used. The completion of this operation will be obvious if the control net is displayed. The user will see the new adapted control net pop up. The new control net should look like the one shown in figure 5. The user now has two choices, is the control net acceptable, or should the user select "restore CN and Grid" to bring back the initial control net? The control net that now exists will capture the strong shock, but may not capture the reflected weak shock adequately.

To capture the weaker reflected shock, the user recalculates the grid with the new control net by selecting the "ReCalculate Grid" entry, and then selects "Redefine BLOCK" entry under the HOME menu, allowing grid adaptation only in the weaker shock region without the effect of the stronger shock. When the "redefine BLOCK" entry is selected, TURBO-AD will instruct the user to select the i_{min}, j_{min} point using the left mouse button and the i_{max}, j_{max} point using the center button. The left button should be used to select a point near the expansion corner. The user can determine what region is selected by the color changes of the grid. The region can be changed over and over while in this option. To exit, the user should press the right mouse button. A new control net will be created and attached to the new local region as shown in figure 6(a). At this time the user should return to the adaptation menu and select "Calculate Sources" entry. This will only calculate sources in the new region as shown in figure 6(b). The user can then use "Adapt CN" and "ReCalculate Grid" to adapt to the weaker shock. The process should be repeated two or three times starting with the calculation of the sources. The resulting grid should look similar to the one shown in figure 7. The user should note that there are discontinuities along the local block boundaries. This will be smoothed out when the user returns to "Redefine BLOCK" and chooses the entire grid and does a grid calculation. The resulting grid is shown in figure 8.

Further improvements can still be made by using local control point manipulations as discussed in reference 6. The final adapted control net, grid, and the new solution are shown in figure 9. The new solution was obtained using the adapted grid from a separate Euler code. The strong shocks have both become crisp and discernable.

CONCLUDING REMARKS

Two technologies are combined to develop an interactive solution-adaptive grid generation program, TURBO-AD. One is the solution-adaptation technique that uses the parametric mapping with control sources that are derived from the flow solutions. The other is the control point form of

algebraic grid generation technique that provides precise local control. The control point form technique is efficient as an algebraic method. Significant computational savings are attained by adapting the control net to solutions instead of the usual direct grid adaptation to solutions. Further savings in the computational time is achieved in TURBO-AD by cutting off the computation of the influence function K at grid points where K decays below a user specified value.

TURBO-AD is being developed on the IRIS 4-D series workstations. It can display the grid, solution contour, and the control net separately or can overlay one over the other for examination. With the improvement in computational speed, two-dimensional solution-adaptive grid generation is now a viable option on graphics workstations. As shown with an illustration, TURBO-AD has global grid control over the entire domain, control over subregions, as well as local grid control capability.

TURBO-AD has had limited testing. Further testing and enhancements are needed to improve robustness and efficiency of the code. Also needed are guidelines for the selection of adaptation parameters.

REFERENCES

1. Lee, K.D.; Loellbach, J.M.; and Pierce, T.R.: Solution Adaptive Grid Generation Using a Parametric Mapping. Numerical Grid Generation in Computational Fluid Mechanics '88: Proceedings of the Second International Conference, S. Sengupta, et al., eds., Pineridge Press, Ltd, Swansea, Wales, 1988, pp. 455-464.
2. Lee, K.D.; and Loellbach, J.M.: Geometry-Adaptive Surface Grid Generation Using a Parametric Projection. J. Aircraft, vol. 26, Feb. 1989, pp. 162-167.
3. Lee, K.D.; Henderson, T.L.; and Choo, Y.K.: Grid Quality Improvement by a Grid Adaptation Technique. Numerical Grid Generation in Computational Fluid Dynamics and Related Fields, A.S.-Arcilla, et al., eds., North-Holland, 1991, pp. 597-606.
4. Eiseman, P.R.: A control Point Form of Algebraic Grid Generation. Int. J. Num. Methods Fluids, vol. 8, Oct. 1988, pp. 1165-1181.
5. Eiseman, P.R.; Snyder, A.; Choo, Y.K.: Dynamics of Local Grid Manipulations for Internal Flow Problems, Computational Fluid Dynamics Symposium on Aeropropulsion, NASA CP-10045, pp. 30-26, 1990.
6. Choo, Y.K.; Miller, D.P.; and Reno, C.: Implementation of Control Point form of Algebraic Grid Generation Technique, Numerical Grid Generation in Computational Fluid Dynamics and Related Fields, ed. by A.S.-Arcilla, et al. North-Holland, pp. 331-341, 1991.

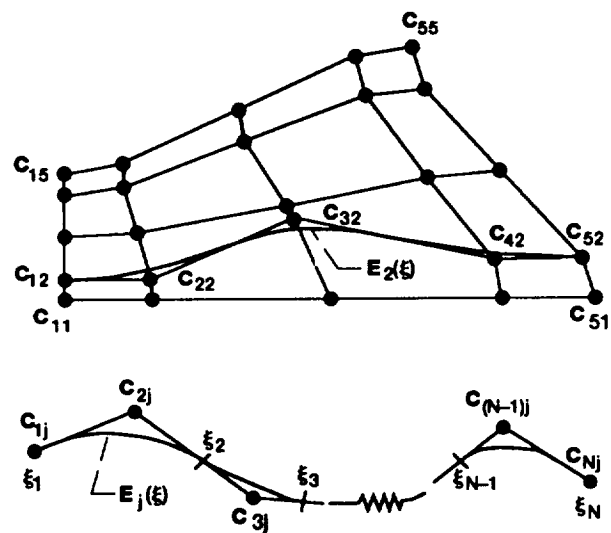
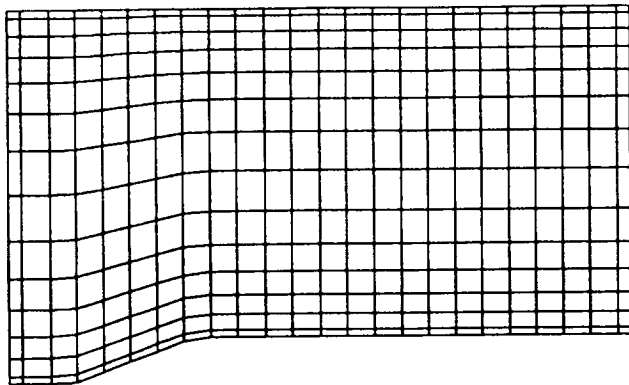
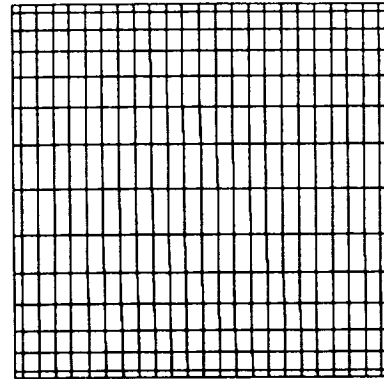


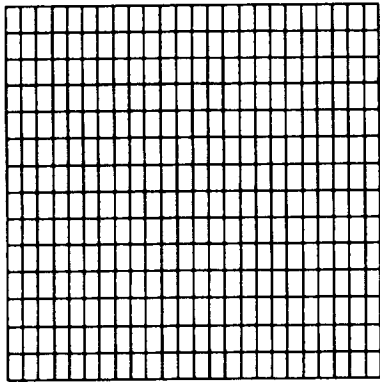
Figure 1.—Control net and construction of a curve.



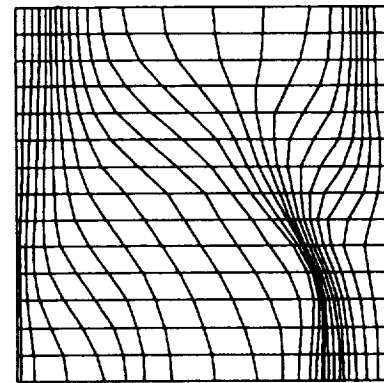
(a) Initial control net in physical space (x, y) .



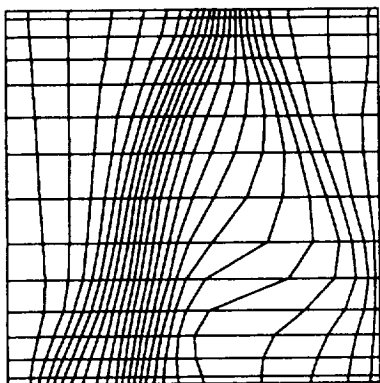
(b) Initial control net in parametric space (s, t) .



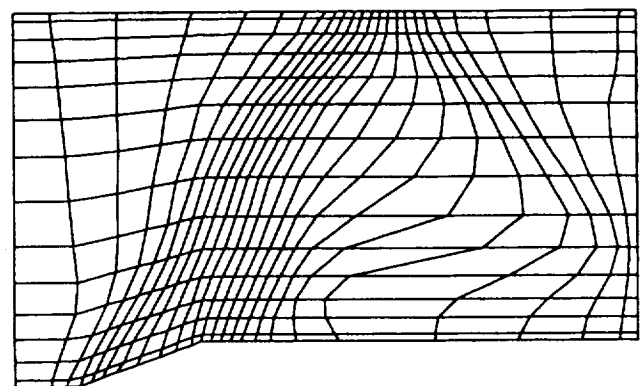
(c) Uniform control net in parametric space (u, v) .



(d) Adapted control net in parametric space (u', v') .

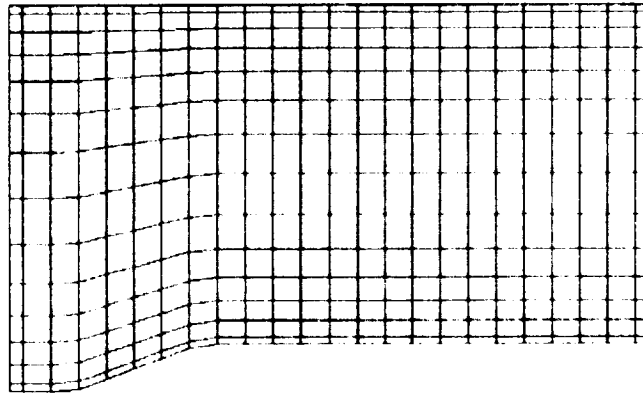


(e) Adapted control net in parametric space (s, t) .

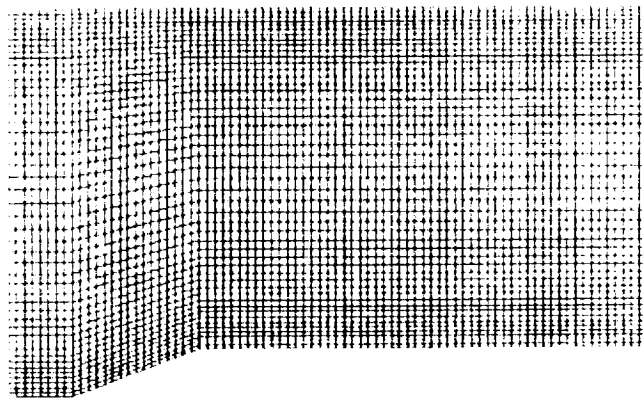


(f) Adapted control net in physical space (x, y) .

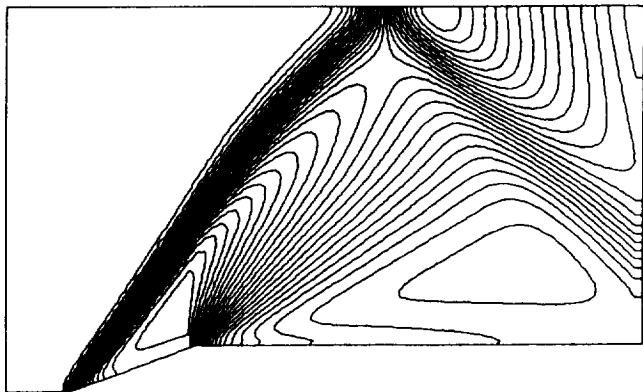
Figure 2.—The adaptation process for the control net.



(a) Control net.



(b) Grid.



(c) Mach number distribution.

Figure 3.—Initial results.

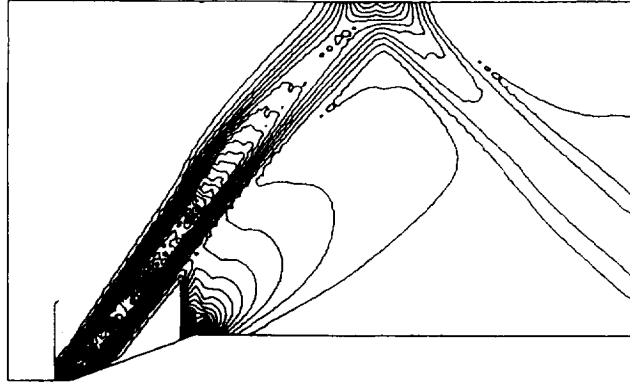


Figure 4.—Source distribution in S-direction.

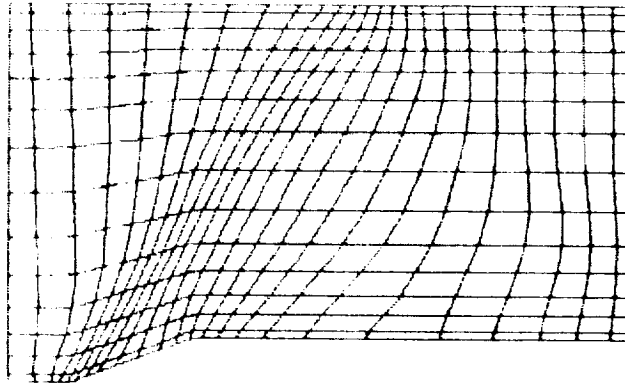
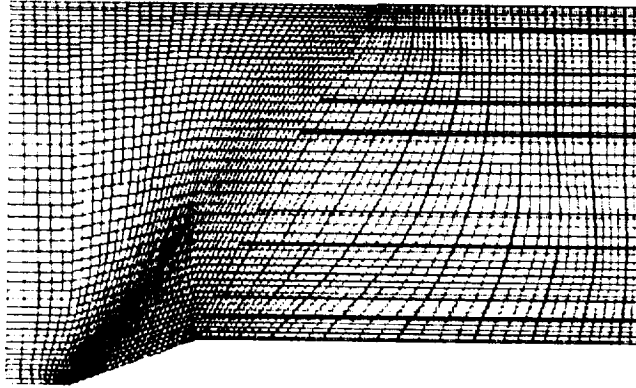
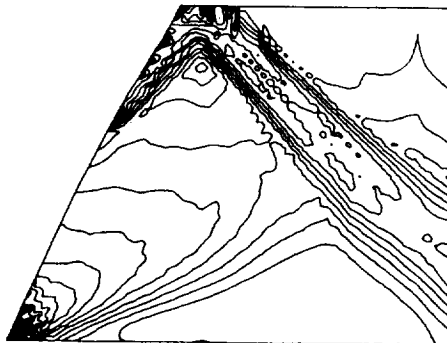


Figure 5.—Adapted control net.



(a) Sub-block control net.



(b) Local source distribution.

Figure 6.—Sub-block used for local adaptation to resolve the weak reflected shock.

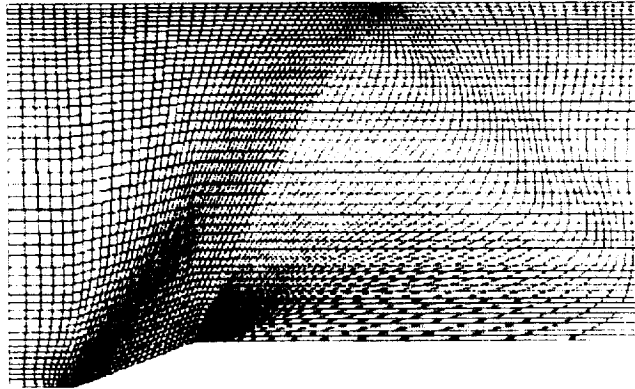


Figure 7.—Solution adapted grid after local adaptation.

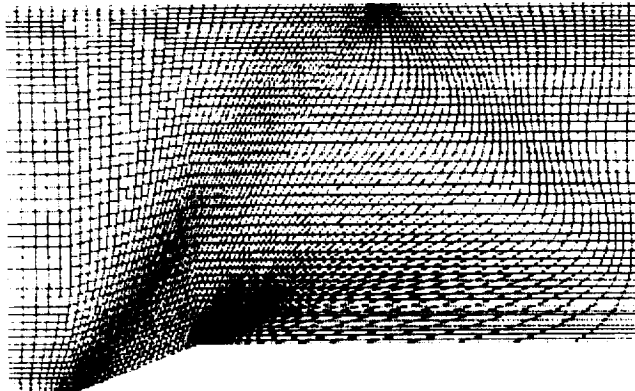
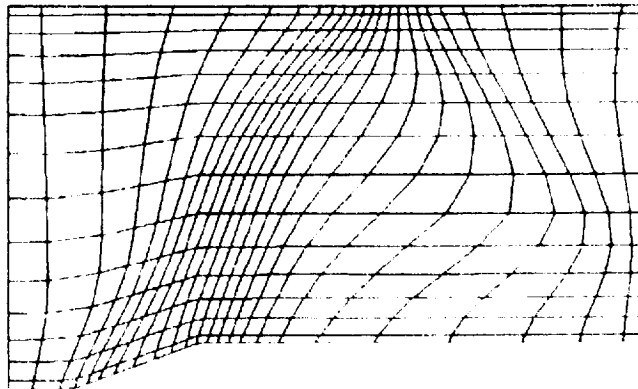
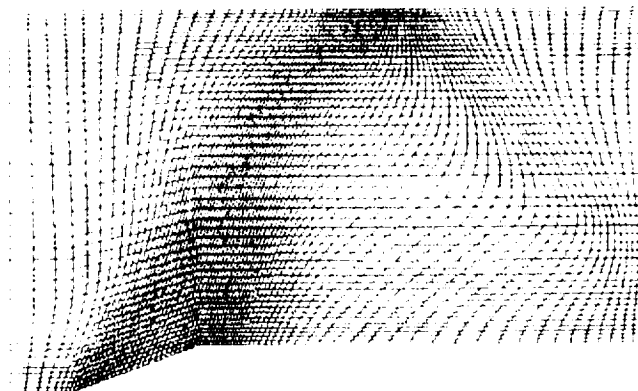


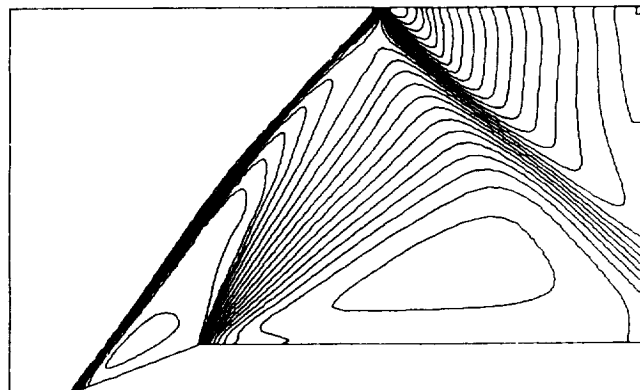
Figure 8.—Solution adapted grid after recalculation of grid using entire domain.



(a) Control net.



(b) Grid.



(c) Mach number distribution.

Figure 9.—Final results.

